MP0390

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appeal No. _______________

Application No.:	10/626,507
Filing Date:	July 24, 2003
Appellants:	Son Ho et al.
Conf. No.:	
Group Art Unit:	2188
Examiner:	Kaushikkumar Patel
Title:	LINE CACHE CONTROLLER

BRIEF ON APPEAL ON BEHALF OF APPELLANTS AND PETITION FOR EXTENSION OF TIME

Mail Stop Appeal Brief-Patents
P.O. Box 1450
Alexandria, VA 22313-1450

March 6, 2007

Sir:

This appeal is from the decision of the Patent Examiner dated May 2, 2007, rejecting claims 1-28, 44-63, and 79-105, which are reproduced in Appendix A of this Appeal Brief.

Applicant hereby petitions under the provisions of 37 C.F.R. § 1.136(a) for an extension of time in which to respond to the outstanding Office Action and includes a fee as set forth in 37 C.F.R. § 1.17(a) with this response for such extension of time.

TABLE OF CONTENTS

Application No.: 10/626,507 1
I. REAL PARTY IN INTEREST 3
II. RELATED APPEALS AND INTERFERENCES 3
III. STATUS OF THE CLAIMS 3
IV. STATUS OF THE AMENDMENTS 3
V. SUMMARY OF THE CLAIMED SUBJECT MATTER 4
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 8
VII. ARGUMENTS 9
A. The Rejections 9
B. Claim Distinctions 10
1. Distinctions regarding independent Claims 1, 12, 16, 19, 21, 44, 51, 55, 58, 59, 79, 90, 94, 96, and 98 10
2. Dependent Claims 2-15, 17-18, 20, 22-28, 45-50, 52-54, 56-57, 60-63, 80-89, 91-93, 95, 97, and 99-105 14
VIII. CONCLUSION 14
IX. APPENDIX A 16
CLAIMS APPENDED 16
X. APPENDIX B 40
EVIDENCE APPENDED 40
XI. APPENDIX C 40
RELATED PROCEEDINGS APPENDED 40

BRIEF ON APPEAL ON BEHALF OF APPELLANTS

In support of the Notice of Appeal filed August 23, 2007, appealing the Examiner's Rejection of each of claims 1-28, 44-63, and 79-105, mailed May 2, 2007, which appear in the attached Appendix A, Appellants hereby provide the following remarks.

I. REAL PARTY IN INTEREST

The present application is assigned to Marvell International, Ltd as recorded in the Patent and Trademark Office at Reel 014998, Frame 0878 and Reel 014998, Frame 0797.

II. RELATED APPEALS AND INTERFERENCES

An appeal is pending in related Patent Application No. 10/646,289. The undersigned, the Assignee, and the Appellants do not know of any other appeals or interferences which would directly affect or that would be directly affected by, or have a bearing on, the Board's decision in this Appeal.

III. STATUS OF THE CLAIMS

Claims 1-28, 44-63, and 79-105 are reproduced in the attached Appendix A and are the claims on Appeal. Each of these claims stands rejected.

IV. STATUS OF THE AMENDMENTS

There are no pending amendments filed subsequent to the final rejection.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Independent claim 1 recites a line cache control system (see FIG. 1) that controls data flow between a line cache (see FIG. 1, element 58; Page 7, Line 1), a first central processing unit (CPU) (see FIG. 1, element 50, Page 6, Lines 20-21) and first and second memory devices (see FIG. 1, elements 66 and 70; Page 7, Lines 1-3). The line cache control system includes a first line cache interface (see FIG. 1, element 52; Page 6, Line 21) that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request. A first memory interface (see FIG. 1, element 78; Page 7, Lines 4-5) communicates with the first memory device. A second memory interface (see FIG. 1, element 80; Page 7, Lines 10-11) communicates with the second memory device. A line cache (see FIG. 1, element 58; Page 7, Line 1) receives a second address that is based on the first address and includes a memory select portion (see FIG. 3, elements 134 and 134'; Page 10, Lines 8-22). A switch (see FIG. 1, element 64; Page 7, Lines 1-3) receives the second address and that selectively connects said line cache to one of said first and second memory interfaces based on the memory select portion.

The line cache receives said second address, said line cache compares said second address to stored addresses in said line cache, returns data associated with said second address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs. The switch includes a plurality of selectors (see FIG. 6, elements 244, 246, and 248; Page 14, Lines 3-14) that each receive the second address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address.

Independent claim 12 further recites a second CPU (see FIG. 2, element 50-2; Page 8, Lines 4-6). The line cache includes line cache memory (see FIG. 4, element 58; Page 11, Lines 7-9) that stores data, a content addressable memory (CAM) (see FIG. 4, element 154; Page 11, Lines 7-9) that stores addresses associated with said data stored in said line cache memory, and a line cache module that includes a line cache state machine (see FIGS. 6 and 13, element 234; Page 17, Line 17-Page 19,

Line 7) that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs. The line cache memory includes multiple pages and said line cache module allows one page to be accessed by one of the first CPU and said second CPU while the other of the first CPU and said second CPU is waiting for data retrieval in another page (see Page 12, Lines 14-18).

Independent claim 16 further recites a least used page device (see FIG. 12, element 330; Page 17, Lines 8-16) that identifies a least used page in said line cache, that replaces said least used page with data retrieved from one of the first and second memory devices when a miss occurs, and that identifies a second least used page and wherein said line cache state module checks internal states of the first CPU.

Independent claim 19 further recites that the first CPU executes an application and that said line cache has a line width and number of pages that are based on said application (see Page 6, Lines 12-14).

Independent claim 21 further recites a second line cache interface (see FIG. 2, element 52-2; Page 8, Lines 4-6) that is associated with the second CPU, that receives a second program read request from the second CPU and that generates a second address from said second program read request. A line cache arbitration device (see FIG. 2, element 100; Page 8, Lines 9-11) communicates with said first and second line cache interfaces and said line cache and resolves line cache access conflicts between the first CPU and the second CPU, and generates a translated address based on one of the first address and the second address that includes a memory select portion.

Independent claim 44 recites a method for operating a line cache (see FIG. 1, element 58; Page 7, Line 1). The method includes receiving a first program read request from a first CPU (see FIG. 1, element 50, Page 6, Lines 20-21) at a first line cache interface (see FIG. 1, element 52; Page 6, Line 21), generating a first address from said first program read request, generating a translated address that includes a memory select portion (see FIG. 3, elements 134 and 134'; Page 10, Lines 8-22) based on said first address, selectively connecting (see FIG. 1, element 64; Page 7, Lines 1-3) said line cache to one of first (see FIG. 1, element 78; Page 7, Lines 4-5) and second (see FIG. 1, element 80; Page 7, Lines 10-11) memory interfaces for first and second

memory devices (see FIG. 1, elements 66 and 70; Page 7, Lines 1-3), respectively, based on said memory select portion, wherein said selectively connecting includes selectively connecting said line cache to said one of said first and second memory interfaces with a plurality of selectors (see FIG. 6, elements 244, 246, and 248; Page 14, Lines 3-14) that each receive the translated address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the translated address, comparing said first address to stored addresses in said line cache when said line cache receives said first address, returning data associated with said first address if a match occurs, and retrieving data from one of said first and second memory devices if a miss occurs (see FIG. 3, elements 134 and 134'; Page 10, Lines 8-22).

Independent claim 51 further recites receiving a second program read request from a second CPU (see FIG. 2, element 50-2; Page 8, Lines 4-6) at a second line cache interface (see FIG. 2, element 52-2; Page 8, Lines 4-6), generating a second address from said second program read request, resolving line cache access conflicts between said first and second CPUs (see FIG. 2, element 100; Page 8, Lines 9-11), storing data in line cache memory (see FIG. 4, element 58; Page 11, Lines 7-9), storing addresses associated with said data stored in said line cache memory (see FIG. 4, element 154; Page 11, Lines 7-9), determining when one of a hit and a miss occurs, and managing retrieval of data from said first and second memory devices when said miss occurs (see FIGS. 6 and 13, element 234; Page 17, Line 17-Page 19, Line 7). The line cache memory includes multiple pages and further comprising allowing one page to be accessed by one of said first CPU and said second CPU while the other of said first CPU and said second CPU is waiting for data retrieval in another page (see Page 12, Lines 14-18).

Independent claim 55 further recites identifying a first least used page in said line cache, identifying a second least used page, and checking at least one internal state of said first CPU (see FIG. 12, element 330; Page 17, Lines 8-16).

Independent claim 58 further recites that said first CPU executes an application and basing a line width and number of pages of said line cache on said application (see Page 6, Lines 12-14).

Independent claim 59 further recites resolving line cache access conflicts between said first CPU and said second CPU (see FIG. 2, element 100; Page 8, Lines 9-11).

Independent claim 79 recites a line cache control system (see FIG. 1) that includes first processing means (see FIG. 1, element 50, Page 6, Lines 20-21) for processing data, first and second memory means for storing data (see FIG. 1, elements 66 and 70; Page 7, Lines 1-3), first line cache interface means (see FIG. 1, element 52; Page 6, Line 21) that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request, first memory interface means (see FIG. 1, element 78; Page 7, Lines 4-5) for communicating with said first memory means, second memory interface means (see FIG. 1, element 80; Page 7, Lines 10-11) for communicating with said second memory means, line cache means (see FIG. 1, element 58; Page 7, Line 1) for storing data and receiving a translated address based on the first address that includes a memory select portion (see FIG. 3, elements 134 and 134'; Page 10, Lines 8-22), and selecting means (see FIG. 1, element 64; Page 7, Lines 1-3) for receiving the translated address and selectively connecting said line cache means to one of said first and second memory interface means based on the memory select portion.

The line cache means receives said translated address, said line cache means compares said translated address to stored addresses in said line cache means, returns data associated with said translated address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs. The selecting means includes a plurality of selectors (see FIG. 6, elements 244, 246, and 248; Page 14, Lines 3-14) that each receive the translated address and each select between first and second sets of signals relating to the first and second memory means, respectively, based on the translated address.

Independent claim 90 further recites data storing means (see FIG. 4, element 58; Page 11, Lines 7-9) for storing data, content addressable memory means (see FIG. 4, element 154; Page 11, Lines 7-9) for storing addresses associated with said data stored in said data storing means, and line cache control means (see FIGS. 6 and 13, element

234; Page 17, Line 17-Page 19, Line 7) that determines when one of a hit and a miss occurs and that manages retrieval of data from said first and second memory means when said miss occurs. The line cache means includes multiple pages and said line cache control means allows one page to be accessed by one of said first and second processing means while the other of said first and second processing means is waiting for data retrieval in another page (see Page 12, Lines 14-18).

Independent claim 94 further recites least used page means (see FIG. 12, element 330; Page 17, Lines 8-16) for identifying a first least used page in said line cache means, said least used page means identifies a second least used page and wherein said line cache control means checks internal states of said first processing means, and said first least used page is replaced when a miss occurs and internal states of said first processing means do not indicate a likelihood that said first least used page will be needed within a predetermined period.

Independent claim further 96 recites that said first processing means executes an application and wherein said line cache has a line width and number of pages that are based on said application (see Page 6, Lines 12-14).

Independent claim 98 further recites second line cache interface means (see FIG. 2, element 52-2; Page 8, Lines 4-6) that is associated with the second CPU, for receiving a second program read request from the second CPU and for generating a second address from said second program read request, and line cache arbitration means (see FIG. 2, element 100; Page 8, Lines 9-11) that communicates with said first and second line cache interface means and said line cache for resolving line cache access conflicts between the first CPU and the second CPU and for generating a translated address based on one of the first address and the second address that includes a memory select portion.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellants seek the Board's review of the rejection of:

(a) Claims 1, 44, and 79 under 35 U.S.C. § 103(a) as being unpatentable over Zaidi et al. (U.S. Pat. No. 6,601,126 B1) and Jim Handy (The Cache Memory Book, second edition, published 1998) and Taylor et al. (U.S. Pat. No. 5,699,551), in further view of Jeddeloh (U.S. Pat. No. 7,133,972).

(b) Claims 16, 55, and 94 under 35 U.S.C. § 103(a) as being unpatentable over Zaidi et al. (U.S. Pat. No. 6,601,126 B1), Jim Handy (The Cache Memory Book, second edition, published 1998), Jeddeloh (U.S. Pat. No. 7,133,972), and Taylor et al. (U.S. Pat. No. 5,699,551) in further view of Bryant et al. (U.S. Pat. No. 4,008,460).

(c) Claims 19, 58, and 96 under 35 U.S.C. § 103(a) as being unpatentable over Zaidi et al. (U.S. Pat. No. 6,601,126 B1), Jim Handy (The Cache Memory Book, second edition, published 1998),Taylor et al. (U.S. Pat. No. 5,699,551), Jeddeloh (U.S. Pat. No. 7,133,972) and Barroso et al. (U.S. Pat. No. 6,725,334 B2) in further view of Veidenbaum et al. (Adapting Cache Line Size to Application Behavior, pub. 1999).

(d) Claims 12, 51, and 90 under 35 U.S.C. § 103(a) as being unpatentable over Zaidi et al. (U.S. Pat. No. 6,601,126 B1), Jim Handy (The Cache Memory Book, second edition, published 1998),Taylor et al. (U.S. Pat. No. 5,699,551), Jeddeloh (U.S. Pat. No. 7,133,972) and Barroso et al. (U.S. Pat. No. 6,725,334 B2) in further view of Ebner et al. (U.S. Pat. No. 6,928,525).

(e) Claims 21, 59, and 98 under 35 U.S.C. § 103(a) as being unpatentable over Zaidi et al. (U.S. Pat. No. 6,601,126), Jim Handy (The Cache Memory Book, second edition, published 1998), Taylor et al. (U.S. Pat. No. 5,699,551) and Jeddeloh (U.S. Pat. No. 7,133,972) as applied to claims 1-5, 11, 13-15, 44-48 and 79-83 above, and further in view of Barroso et al. (U.S. Pat. No.6,725,334).

VII. ARGUMENTS

A. The Rejections

The rejections that are the subject of this appeal are: a rejection of each of independent claims 1, 12, 16, 19, 21, 44, 51, 55, 58, 59, 79, 90, 94, 96, and 98 under 35

U.S.C. § 103(a) as allegedly being unpatentable over Zaidi in view of one or more of Jim Handy, Taylor, Bryant, Jeddeloh, Barroso, and Ebner.

With respect to independent claims 1, 12, 16, 19, 21, 44, 51, 55, 58, 59, 79, 90, 94, 96, and 98, the Examiner admits that Zaidi, Jim Handy, and Taylor explicitly fail to teach the limitation "wherein said switch includes a plurality of selectors that each receive the second address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address." Instead, the Examiner alleges that "the selection of respective signals are inherent in the system of Zaidi, because it is well known...that the signals (such as clock and request and acknowledgement etc.) are required for proper communication to occur between the processor and cache or between cache and lower level storage systems." (See Page 5, Line 18 through Page 6, Line 5 of the Office Action mailed May 2, 2007, citing Jeddeloh, Col. 4, Lines 8-29 and 56-67).

B. Claim Distinctions

1. Distinctions regarding independent Claims 1, 12, 16, 19, 21, 44, 51, 55, 58, 59, 79, 90, 94, 96, and 98

With respect to claim 1, Appellants submit that Zaidi, either singly or in combination with any of the other cited prior art references, fails to show, teach, or suggest at least that said switch includes a plurality of selectors that each receive the second address and **each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address**, and further submit that this structure is not inherent in the other cited prior art references.

As shown in exemplary embodiments in FIGS. 6 and 7 of the present application, a switch 268 includes a first multiplexer/selector 244 that receives a memory select signal lc_addr[24] and selects between a buffer clock signal bf_clk and a flash clock signal f_clk accordingly. A second multiplexer/selector 246 receives the memory select signal lc_addr[24] and selectively outputs a signal to a buffer interface or a flash

interface. A third multiplexer/selector 248 receives the memory select signal lc_addr[24] and selects between buffer and flash acknowledgment signals:

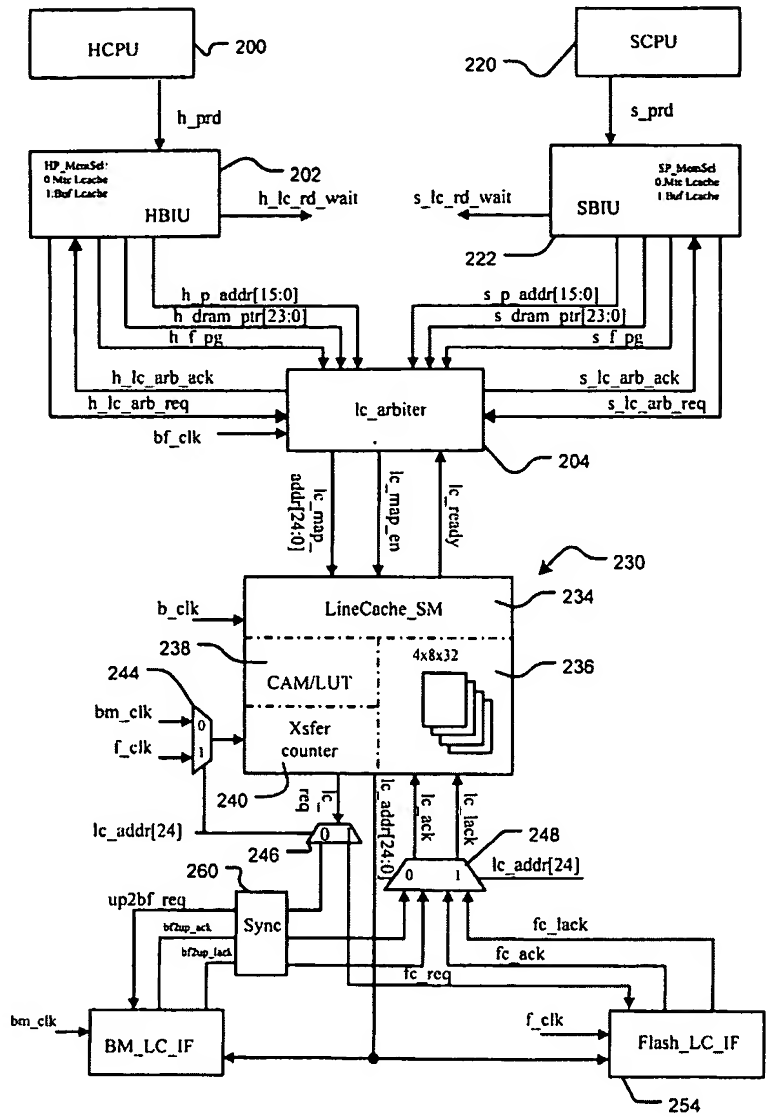


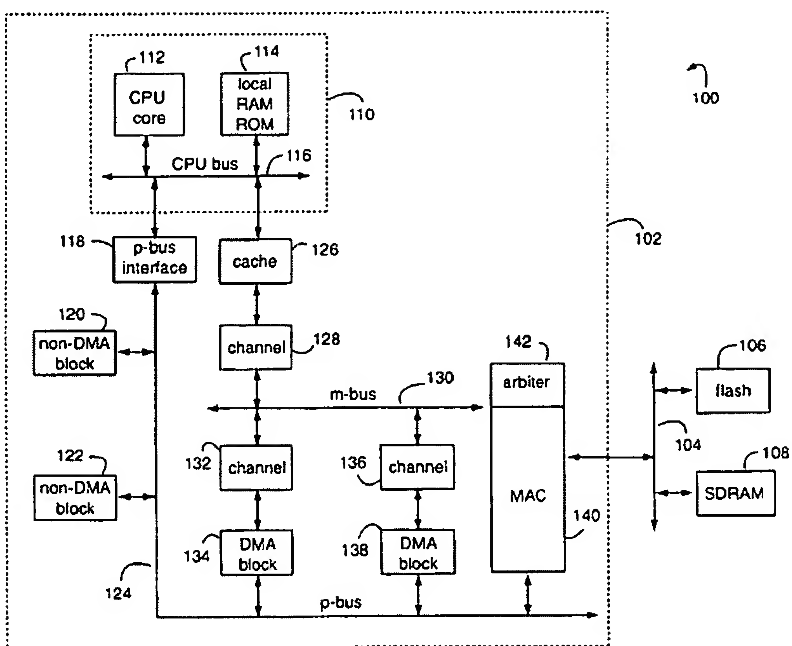
FIG. 6

In other words, the switch includes a plurality of selectors that each select between first and second sets of signals relating to the first and second memory devices based on the memory select signal (e.g. based on the second address). More specifically, each of the selectors (e.g. multiplexers 244, 246, and 248) individually receives the memory select signal lc_addr[24]. The memory select signal includes an

address (i.e. the recited second address) to differentiate between signals associated with different memories.

As best understood by Appellants, Zaidi fails to disclose this limitation. The Examiner **acknowledges that neither Zaidi nor any other cited reference explicitly discloses this limitation, and instead alleges that this structure is inherent**. Appellants respectfully submit that the Examiner has failed to support a prima facie case for inherency.

The Examiner admits that the Zaidi, Jim Handy, and Taylor references are silent as to a plurality of selectors that each select between the recited first and second sets of signals based on the second address. Instead, the Examiner notes that Zaidi teaches CPUs connected to either SRAM or flash memory through a switched channel memory controller, and the selection of respective signals is inherent. (See Page 5, Line 18 through Page 6, Line 5 of the Office Action mailed May 2, 2007). Appellants respectfully note that the Examiner relies on a MAC 140 to disclose the claimed switch and the plurality of selectors:



Appellants respectfully submit that a MAC 140 does not necessarily include a plurality of selectors as Appellants' claims recite. More specifically, Appellant notes that the claims recite that the cache control system includes a switch that receives the

second address and that selectively connects said line cache to one of the first and second memory interfaces based on the memory select portion, and **the switch includes the plurality of selectors**. The Examiner relies on the MAC 140 to disclose the switch. Consequently, the MAC 140 must also include a plurality of selectors that each receive the second address and select between sets of signals based on the second address. Neither Zaidi nor any other cited prior art reference appears to suggest that the MAC 140 would include such a structure.

Appellants note that the fact that a certain characteristic **may occur or be present** in the prior art reference is not sufficient to establish inherency of that characteristic. ***In re Rijckaert***, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) (emphasis added). The Federal Circuit has clearly stated that:

> To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is **necessarily** present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities.'

In re Robertson, 49 USPPQ2d 1949, 1950-1951 (Fed. Cir. 1999) (emphasis added).

"In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic **necessarily** flows from the teachings of the applied prior art." ***Ex Parte Levy***, 17 USPQ2d 1461 (Bd. Pat. App. & Inter. 1990) (emphasis original). Therefore, in order to maintain a valid inherency argument it must be shown that the MAC 140 including the plurality of selectors must **necessarily** flow from the teachings of the Zaidi reference. Appellants respectfully assert that this is not the case here. In particular, Appellants note that a MAC does not necessarily include a plurality of selectors, associated with a switch that selectively connects a line cache to one of first and second memory interfaces based on a memory select portion, arranged as recited in the claims, and the Examiner fails to provide any reference to support this allegation.

The Examiner further notes that Jeddeloh discloses that a switch 160 "can be a set of multiplexers." (Page 6, Lines 5-10 of the Office Action). Appellants respectfully note that the claims recite that the plurality of selectors **each receive the address and**

select between first and second sets of signals relating to first and second memory devices based on the address, and a mere reference to a set of multiplexers fails to disclose the specific structure of this limitation. The Examiner provides no evidence that the switch 160 of Jeddeloh includes a plurality of selectors that select between sets of signals based on a second address (i.e. an address in a memory select portion of an address signal).

Therefore, Appellants respectfully assert that the Examiner has failed to properly support his rejection under either 35 U.S.C. §103 and/or 35 U.S.C. §103 with an inherency argument. Appellants respectfully submit that Zaidi, either singly or in combination with Jeddeloh or any other cited reference, fails to disclose a plurality of selectors that each receive the address and select between first and second sets of signals relating to first and second memory devices based on the address. Appellants further submit that Examiner has failed to support his allegation that this structure is inherent in either Zaidi or Jeddeloh. Accordingly, Appellants respectfully submit that claim 1, as well as its dependent claims, are in condition for allowance for at least the above reasons. Claims 12, 16, 19, 21, 44, 51, 55, 58, 59, 79, 90, 94, 96, and 98, as well as their corresponding dependent claims, are in condition for allowance for at least similar reasons.

2. Dependent Claims 2-15, 17-18, 20, 22-28, 45-50, 52-54, 56-57, 60-63, 80-89, 91-93, 95, 97, and 99-105

With regard to claims 2-15, 17-18, 20, 22-28, 45-50, 52-54, 56-57, 60-63, 80-89, 91-93, 95, 97, and 99-105, these claims are allowable for at least the reasons previously presented with regard to claims 1, 16, 19, 21, 44, 51, 55, 58, 59, 79, 90, 94, 96, and 98, respectively. Accordingly, it is respectfully requested that the rejection of these claims be overturned.

VIII. CONCLUSION

Appellants respectfully request the Honorable Board of Patent Appeals and Interferences to reverse the Examiner's rejection of each of pending claims 1-28, 44-63,

and 79-105. Appellants respectfully submit that the prior art does not teach or suggest one or more limitations of the claims as discussed above. Accordingly, for at least the aforementioned reasons, Appellants respectfully request the Honorable members of the Board of Patent Appeals and Interferences to reverse the outstanding rejections in connection with the present application and permit each of claims 1-28, 44-63, and 79-105 to be passed to allowance in connection with the present application.

Should there be any outstanding matters that need to be resolved in the present application, the Examiner is respectfully requested to contact Michael D. Wiggins, Reg. No. 34,754, or Damian M. Aquino, Reg. No. 54,964, at the telephone number of the undersigned below.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 08-0750 for any additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. § 1.17; particularly, extension of time fees.

Respectfully submitted,

HARNESS, DICKEY, & PIERCE, P.L.C.

By:
Michael D. Wiggins
Reg. No. 34,754
Damian M. Aquino
Reg. No. 54,964

MDW/DMA/dms

<u>Please address all correspondence to</u>:
Harness, Dickey & Pierce, P.L.C.
5445 Corporate Drive
Suite 200
Troy, MI 48098
Customer No. 26703
Tel. No. (248) 641-1600
Fax. No. (248) 641-0270

IX. APPENDIX A

CLAIMS APPENDED

This is a complete and current listing of the claims.

1. (Previously presented) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU) and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache that receives a second address that is based on the first address and includes a memory select portion; and

a switch that receives the second address and that selectively connects said line cache to one of said first and second memory interfaces based on the memory select portion,

wherein when said line cache receives said second address, said line cache compares said second address to stored addresses in said line cache, returns data associated with said second address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs, and

wherein said switch includes a plurality of selectors that each receive the second address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address.

2. (Previously Presented) The line cache control system of claim 1 wherein said first memory device is random access memory (RAM).

3. (Previously Presented) The line cache control system of claim 2 wherein said RAM is one of dynamic RAM (DRAM), synchronous DRAM (SDRAM), and double data rate SDRAM (DDRAM).

4. (Original) The line cache control system of claim 1 wherein said second memory device is flash memory.

5. (Original) The line cache control system of claim 1 wherein said first CPU is an advanced risc machine (ARM) processor.

6. (Previously presented) The line cache control system of claim 1 further comprising:

a second CPU;

a second line cache interface that is associated with said second CPU, that receives a second program read request from said second CPU and that generates a third address from said second program read request; and

a line cache arbitration device that communicates with said first and second line cache interfaces and said line cache and that resolves line cache access conflicts between the first CPU and said second CPU.

7. (Original) The line cache control system of claim 6 further comprising:

a first direct interface that is associated with the first CPU,

wherein said first memory interface includes a second direct interface that communicates with said first direct interface and wherein said first and second direct interfaces allow the first CPU to at least one of read and write data directly to the first memory device.

8. (Original) The line cache control system of claim 7 further comprising:

a third direct interface that is associated with the second CPU,

wherein said second memory interface includes a fourth direct interface that

communicates with said third direct interface and wherein said third and fourth direct interfaces allow said second CPU to at least one of read and write data directly to the second memory device.

9. (Original) The line cache control system of claim 8 further comprising a direct read/write arbitration device that resolves direct memory access conflicts between said first and third direct interfaces.

10. (Original) The line cache control system of claim 6 wherein the first CPU is a host processor for a hard disk drive and said second CPU is a servo processor for said hard disk drive.

11. (Original) The line cache control system of claim 1 wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that includes a line cache state machine that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs.

12. (Previously presented) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU), a second CPU, and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache; and

a switch that selectively connects said line cache to one of said first and second memory interfaces, wherein when said line cache receives said first address, said line cache compares said first address to stored addresses in said line cache, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs,

wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that includes a line cache state machine that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs; and

wherein said line cache memory includes multiple pages and wherein said line cache module allows one page to be accessed by one of the first CPU and said second CPU while the other of the first CPU and said second CPU is waiting for data retrieval in another page,

wherein said switch includes a plurality of selectors that each receive a second address associated with the first address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address.

13. (Original) The line cache control system of claim 11 further comprising a least used page device that identifies a least used page in said line cache.

14. (Original) The line cache control system of claim 13 wherein said least used page device replaces said least used page with data retrieved from one of the first and second memory devices when a miss occurs.

15. (Previously Presented) The line cache control system of claim 11 wherein state transitions of said line cache state machine are based, in part, on at least one internal state of the first CPU.

16. (Previously presented) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU) and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache;

a switch that selectively connects said line cache to one of said first and second memory interfaces, wherein when said line cache receives said first address, said line cache compares said first address to stored addresses in said line cache, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs;

a least used page device that identifies a least used page in said line cache, that replaces said least used page with data retrieved from one of the first and second memory devices when a miss occurs, and that identifies a second least used page and wherein said line cache state module checks internal states of the first CPU,

wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that includes a line cache state machine that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs,

wherein said switch includes a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address.

17. (Previously Presented) The line cache control system of claim 16 wherein said least used page is replaced when a miss occurs and internal states of the first CPU do not indicate a likelihood that said least used page will be needed within a predetermined period.

18. (Previously Presented) The line cache control system of claim 17 wherein said second least used page is replaced when a miss occurs and internal states of the first CPU indicate a likelihood that said least used page will be needed within a predetermined period.

19. (Previously presented) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU) and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache; and

a switch that selectively connects said line cache to one of said first and second memory interfaces, wherein when said line cache receives said first address, said line cache compares said first address to stored addresses in said line cache, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs,

wherein said first CPU executes an application and wherein said line cache has a line width and number of pages that are based on said application,

wherein said switch includes a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address.

20. (Previously Presented) The line cache control system of claim 1 wherein said line cache includes 4 pages of 8 x 32 bits.

21. (Previously presented) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a second line cache interface that is associated with the second CPU, that receives a second program read request from the second CPU and that generates a second address from said second program read request;

a line cache arbitration device that communicates with said first and second line cache interfaces and said line cache that resolves line cache access conflicts between the first CPU and the second CPU, and that generates a translated address based on one of the first address and the second address that includes a memory select portion; and

a switch that receives the translated address and that selectively connects said line cache to one of said first and second memory devices based on the memory select portion,

wherein said switch includes a plurality of selectors that each receive the translated address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the translated address.

22. (Previously Presented) The memory control system of Claim 21 wherein said line cache arbitration device selects one of said first and second addresses and further comprising:

a first memory interface that communicates with the first memory device; and

a second memory interface that communicates with the second memory device, wherein when said line cache receives said selected one of said first and

second addresses from said line cache arbitration device, said line cache compares said selected one of said first and second addresses to stored addresses in said line cache, returns data associated with said selected one of said first and second addresses if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs.

23. (Previously Presented) The memory control system of claim 21 wherein the first memory device is random access memory (RAM) and the second memory device is flash memory.

24. (Previously Presented) The memory control system of claim 21 wherein the first CPU is an embedded processor.

25. (Previously Presented) The memory control system of claim 21 further comprising:

a first direct interface that is associated with the first CPU, wherein said first memory interface includes a second direct interface that communicates with said first direct interface and wherein said first and second direct interfaces allow the first CPU to at least one of read and write data directly to the first memory device; and

a third direct interface that is associated with the second CPU, wherein said second memory interface includes a fourth direct interface that communicates with said third direct interface and wherein said third and fourth direct interfaces allow the second CPU to at least one of read and write data directly to the second memory device.

26. (Previously Presented) The memory control system of claim 25 further comprising a direct read/write arbitration device that resolves direct memory access conflicts between said first and third direct interfaces.

27. (Previously Presented) The memory control system of claim 21 wherein the first CPU is a host processor for a hard disk drive and the second CPU is a servo processor for said hard disk drive.

28. (Previously Presented) The memory control system of claim 21 wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs.

29-43 (Cancelled).

44. (Previously presented) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

generating a translated address that includes a memory select portion based on said first address;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively, based on said memory select portion, wherein said selectively connecting includes selectively connecting said line cache to said one of said first and second memory interfaces with a plurality of selectors that each receive the translated address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the translated address;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs; and

retrieving data from one of said first and second memory devices if a miss occurs.

45. (Previously Ppresented) The method of claim 44 wherein said first memory device is random access memory (RAM).

46. (Previously Presented) The method of claim 45 wherein said RAM is one of dynamic RAM (DRAM), synchronous DRAM (SDRAM), and double data rate SDRAM (DDRAM).

47. (Previously Presented) The method of claim 44 wherein said second memory device is flash memory.

48. (Previously Presented) The method of claim 44 wherein said first CPU is an embedded processor.

49. (Previously Presented) The method of claim 44 further comprising:

receiving a second program read request from a second CPU at a second line cache interface;

generating a second address from said second program read request; and

resolving line cache access conflicts between said first and second CPUs.

50. (Previously Presented) The method of claim 44 further comprising:

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs; and

managing retrieval of data from said first and second memory devices when said miss occurs.

51. (Previously presented) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

receiving a second program read request from a second CPU at a second line cache interface;

generating a second address from said second program read request;

resolving line cache access conflicts between said first and second CPUs;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs;

retrieving data from one of said first and second memory devices if a miss occurs;

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs; and

managing retrieval of data from said first and second memory devices when said miss occurs,

wherein said line cache memory includes multiple pages and further comprising allowing one page to be accessed by one of said first CPU and said second CPU while the other of said first CPU and said second CPU is waiting for data retrieval in another page, and

wherein said selectively connecting includes selectively connecting said line cache to one of said first and second memory interfaces with a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address.

52. (Previously Ppresented) The method of claim 50 further comprising identifying a first least used page in said line cache.

53. (Previously Presented) The method of claim 52 further comprising replacing said first least used page with data retrieved from one of the first and second memory devices when a miss occurs.

54. (Previously Ppresented) The method of claim 50 further comprising operating said line cache based on at least one internal state of said first CPU.

55. (Previously presented) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively, wherein said selectively connecting includes selectively connecting said line cache to said one of said first and second memory interfaces with a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs;

retrieving data from one of said first and second memory devices if a miss occurs;

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs;

managing retrieval of data from said first and second memory devices

when said miss occurs;

identifying a first least used page in said line cache;

identifying a second least used page; and

checking at least one internal state of said first CPU.

56. (Previously Presented) The method of claim 55 further comprising replacing said first least used page when a miss occurs and said at least one internal state of said first CPU do not indicate a likelihood that said first least used page will be needed within a predetermined period.

57. (Previously Presented) The method of claim 55 further comprising replacing said second least used page when a miss occurs and said at least one internal state of said first CPU indicate a likelihood that said first least used page will be needed within a predetermined period.

58. (Previously presented) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively, wherein said selectively connecting includes selectively connecting said line cache to said one of said first and second memory interfaces with a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the second address;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs; and

retrieving data from one of said first and second memory devices if a miss occurs,

wherein said first CPU executes an application and further comprising basing a line width and number of pages of said line cache on said application.

59. (Previously presented) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

receiving a second program read request from a second CPU at a second line cache interface;

generating a second address from said second program read request;

generating a translated address based on at least one of the first address and the second address, wherein the translated address includes a memory select portion;

resolving line cache access conflicts between said first CPU and said second CPU; and

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively, based on the memory select portion, wherein said selectively connecting includes selectively connecting said line cache to one of said first and second memory interfaces with a plurality of selectors that each receive the memory select portion and each select between first and second sets of signals relating to the first and second memory devices, respectively, based on the memory select portion.

60. (Previously Presented) The method of Claim 59 further comprising:

selecting one of said first and second addresses;

comparing said selected one of said first and second addresses to stored addresses in said line cache;

returning data associated with said selected one of said first and second addresses if a match occurs; and

retrieving data from one of said first and second memory devices if a miss occurs.

61. (Previously Presented) The method of Claim 59 wherein said first memory device is random access memory (RAM) and said second memory device is flash memory.

62. (Previously Presented) The method of Claim 59 wherein said first CPU is an embedded processor.

63. (Previously Presented) The method of Claim 59 further comprising:

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs; and

managing retrieval of data from said first and second memory devices when said miss occurs.

64-78. (Cancelled)

79. (Previously presented) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data and receiving a translated address based on the first address that includes a memory select portion; and

selecting means for receiving the translated address and selectively connecting said line cache means to one of said first and second memory interface

means based on the memory select portion,

wherein when said line cache means receives said translated address, said line cache means compares said translated address to stored addresses in said line cache means, returns data associated with said translated address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs, and

wherein said selecting means includes a plurality of selectors that each receive the translated address and each select between first and second sets of signals relating to the first and second memory means, respectively, based on the translated address.

80. (Previously Presented) The line cache control system of claim 79 wherein said first memory means is random access memory (RAM).

81. (Previously Presented) The line cache control system of claim 80 wherein said RAM is one of dynamic RAM (DRAM), synchronous DRAM (SDRAM), and double data rate SDRAM (DDRAM).

82. (Previously Presented) The line cache control system of claim 79 wherein said second memory means is flash memory.

83. (Previously Presented) The line cache control system of claim 79 wherein said first processing means is an embedded processor.

84. (Previously Presented) The line cache control system of claim 79 further comprising:

second processing means for processing data;

second line cache interface means that is associated with said second processing means, that receives a second program read request from said second processing means and that generates a second address from said second program read request; and

line cache arbitration means that communicates with said first and second line cache interface means for resolving access conflicts to said line cache means between said first processing means and said second processing means.

85. (Previously Presented) The line cache control system of claim 84 further comprising:

first direct interface means that is associated with said first processing means,

wherein said first memory interface means includes a second direct interface means that communicates with said first direct interface means and wherein said first and second direct interface means allow said first processing means to at least one of read and write data directly to said first memory means.

86. (Previously Presented) The line cache control system of claim 85 further comprising:

third direct interface means that is associated with said second processing means,

wherein said second memory interface means includes a fourth direct interface means that communicates with said third direct interface means and wherein said third and fourth direct interface means allow said second processing means to at least one of read and write data directly to said second memory means.

87. (Previously Presented) The line cache control system of claim 86 further comprising direct read/write arbitration means for resolving direct memory access conflicts between said first and third direct interface means.

88. (Previously Presented) The line cache control system of claim 84 wherein said first processing means is a host processor for a hard disk drive and said second processing means is a servo processor for said hard disk drive.

89. (Previously Presented) The line cache control system of claim 79 wherein

said line cache means includes:

data storing means for storing data;

content addressable memory means for storing addresses associated with said data stored in said data storing means; and

line cache control means that determines when one of a hit and a miss occurs and that manages retrieval of data from said first and second memory means when said miss occurs.

90. (Previously presented) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data that includes:

data storing means for storing data;

content addressable memory means for storing addresses associated with said data stored in said data storing means; and

line cache control means that determines when one of a hit and a miss occurs and that manages retrieval of data from said first and second memory means when said miss occurs; and

selecting means for selectively connecting said line cache means to one of said first and second memory interface means,

wherein when said line cache means receives said first address, said line cache means compares said first address to stored addresses in said line cache means, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs;

wherein said line cache means includes multiple pages and wherein said line cache control means allows one page to be accessed by one of said first and second processing means while the other of said first and second processing means is waiting for data retrieval in another page; and

wherein said selecting means includes a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory means, respectively, based on the second address.

91. (Previously Presented) The line cache control system of claim 89 further comprising least used page means for identifying a first least used page in said line cache means.

92. (Previously Presented) The line cache control system of claim 91 wherein said line cache control means replaces said first least used page with data retrieved from one of the first and second memory means when a miss occurs.

93. (Previously Presented) The line cache control system of claim 92 wherein said least used page means identifies a second least used page and wherein said line cache control means checks internal states of said first processing means.

94. (Previously presented) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data that includes:

data storing means for storing data;

content addressable memory means for storing addresses associated with said data stored in said data storing means; and

line cache control means that determines when one of a hit and a miss occurs and that manages retrieval of data from said first and second memory means when said miss occurs;

selecting means for selectively connecting said line cache means to one of said first and second memory interface means;

least used page means for identifying a first least used page in said line cache means

wherein when said line cache means receives said first address, said line cache means compares said first address to stored addresses in said line cache means, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs, said line cache control means replaces said first least used page with data retrieved from one of the first and second memory means when a miss occurs, said least used page means identifies a second least used page and wherein said line cache control means checks internal states of said first processing means, and said first least used page is replaced when a miss occurs and internal states of said first processing means do not indicate a likelihood that said first least used page will be needed within a predetermined period, and

wherein said selecting means includes a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory means, respectively, based on the second address.

95. (Previously Presented) The line cache control system of claim 94 wherein said second least used page is replaced when a miss occurs and internal states of said first processing means indicate a likelihood that said first least used page will be needed within a predetermined period.

96. (Previously presented) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data; and

selecting means for selectively connecting said line cache means to one of said first and second memory interface means,

wherein when said line cache means receives said first address, said line cache means compares said first address to stored addresses in said line cache means, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs, and said first processing means executes an application and wherein said line cache has a line width and number of pages that are based on said application, and

wherein said selecting means includes a plurality of selectors that each receive a second address based on the first address and each select between first and second sets of signals relating to the first and second memory means, respectively, based on the second address.

97. (Previously Presented) The line cache control system of claim 79 wherein said line cache includes 4 pages of 8 x 32 bits.

98. (Previously presented) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

first line cache interface means that is associated with the first CPU, for

receiving a first program read request from the first CPU and for generating a first address from said first program read request;

second line cache interface means that is associated with the second CPU, for receiving a second program read request from the second CPU and for generating a second address from said second program read request;

line cache arbitration means that communicates with said first and second line cache interface means and said line cache for resolving line cache access conflicts between the first CPU and the second CPU and for generating a translated address based on one of the first address and the second address that includes a memory select portion; and

switching means for receiving the translated address and for selectively connecting said line cache to one of said first and second memory devices based on the memory select portion, wherein said selecting means includes a plurality of selectors that each receive the translated address and each select between first and second sets of signals relating to the first and second memory device, respectively, based on the translated address.

99. (Previously Presented) The memory control system of Claim 98 wherein said line cache arbitration means selects one of said first and second addresses and further comprising:

first memory interface means for communicating with the first memory device; and

second memory interface means for communicating with the second memory device, wherein when said line cache receives said selected one of said first and second addresses from said line cache arbitration means, said line cache compares said selected one of said first and second addresses to stored addresses in said line cache, returns data associated with said selected one of said first and second addresses if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs.

100. (Previously Presented) The memory control system of claim 98 wherein

the first memory device is random access memory (RAM) and the second memory device is flash memory.

101. (Previously Presented) The memory control system of claim 98 wherein the first CPU is an embedded processor.

102. (Previously Presented) The memory control system of claim 98 further comprising:

first direct interface means that is associated with the first CPU, wherein said first memory interface means includes a second direct interface means for communicating with said first direct interface means and wherein said first and second direct interface means allow the first CPU to at least one of read and write data directly to the first memory device; and

a third direct interface means that is associated with the second CPU, wherein said second memory interface means includes a fourth direct interface means for communicating with said third direct interface means and wherein said third and fourth direct interface means allow the second CPU to at least one of read and write data directly to the second memory device.

103. (Previously Presented) The memory control system of claim 102 further comprising direct read/write arbitration means for resolving direct memory access conflicts between said first and third direct interface means.

104. (Previously Presented) The memory control system of claim 98 wherein the first CPU is a host processor for a hard disk drive and the second CPU is a servo processor for said hard disk drive.

105. (Previously Presented) The memory control system of claim 98 wherein said line cache includes:

data storing means for storing data;

content addressable memory (CAM) means for storing addresses

associated with said data stored in said storing means; and

line cache control means for determining when one of a hit and a miss occurs and for managing retrieval of data from the first and second memory devices when said miss occurs.

106-120 (Cancelled).

X. APPENDIX B

EVIDENCE APPENDED

A copy of the Office Action mailed May 2, 2007 is attached.

XI. APPENDIX C

RELATED PROCEEDINGS APPENDED

The undersigned, the Assignee, and the Appellants do not know of any other appeals or interferences which would directly affect or that would be directly affected by, or have a bearing on, the Board's decision in this Appeal.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/626,507	07/24/2003	Son Ho	MP0390	1965

26703 7590 05/02/2007
HARNESS, DICKEY & PIERCE P.L.C.
5445 CORPORATE DRIVE
SUITE 200
TROY, MI 48098

EXAMINER	
PATEL, KAUSHIKKUMAR M	

ART UNIT	PAPER NUMBER
2188	

MAIL DATE	DELIVERY MODE
05/02/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

S0S9-000090

MDW

FINAL OA
DUE: 8-9-07

OIPE
MAR 0 6 2008
PATENT & TRADEMARK OFFICE
IAP39

Office Action Summary	Application No.	Applicant(s)	
	10/626,507	HO ET AL.	
	Examiner	Art Unit	
	Kaushikkumar Patel	2188	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _05 March 2007_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-28, 44-63 and 79-105_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-28,44-63 and 79-105_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _24 July 2003_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _1/4/2007_.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to applicant's communication filed March 05, 2007 in response to PTO office action mailed December 04, 2006. The applicant's remarks and amendments to the claims and/or specification were considered with the results that follow.

2. In response to the last office action, claims 1, 6, 12, 16, 19, 21, 44, 51, 55, 58-59, 79, 90, 94, 96 and 98 have been amended. No claims have been added. Claims 29-43, 64-78 and 106-120 have been canceled. As a result, claims 1-28, 44-63 and 79-105 remain pending in this application.

3. Rejection of claims under 35 USC 112, second paragraph is withdrawn due to amendments filed on March 05, 2007.

4. Double Patenting rejection of claims with co-pending application 10/646289 have been withdrawn due to terminal disclaimer filed in later filed application (10/646289) on March 05, 2007.

Response to Arguments

5. Applicant's arguments with respect to claims 1-120 have been considered but are moot in view of the new ground(s) of rejection.

Information Disclosure Statement

6. The information disclosure statement (IDS) submitted on January 04, 2007 had considered by the examiner.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-5, 11,13-15, 20, 44-48, 50, 52-54, 79-83, 89, 91-92, and 97 are rejected under 35 U.S.C. 103(a) as being unpatentable in view of Zaidi et al. (US 6,601,126 B1) (Zaidi herein after), Jim Handy (The Cache Memory Book, second edition, published 1998) (Jim herein after), Taylor et al. (5,699,551) (Taylor herein after) and Jeddeloh (US 7,133,972).

As per claims 1, 44, 50 and 79, Zaidi teaches a cache control system (fig. 1) that controls data flow between a line cache (fig, 1, item 126), a first central processing unit (CPU) (fig. 1, item 110) and first and second memory devices (fig. 1, items 106 and 108), comprising:

a first line cache interface that is associated with the first CPU (taught as cache and channel controller interface the CPU bus, column 4, lines 39-41), that receives a

first program read request from CPU and that generates a first address from said first program read request (column 23, lines 30-32);

a first memory interface that communicate with first memory device and second memory interface that communicates with the second memory device (figs. 21-23, two memory devices, flash and SDRAM are connected to memory bus through MAC, which teaches first and second interfaces connected to first and second memory devices);

a cache that receives address that includes memory select portion; and a switch that selectively connects said line cache to one of said first and second memory interfaces (fig. 1, item 126 is a cache and even though it is explicitly not taught, when CPUs are provided with cache, CPUs initially try to access data from cache, thus cache receives address, column 23, lines 31-34 and lines 41-45, taught as CPUs supply a request and an address, the <u>address includes both the port, device or memory bank address [memory select portion] and the requested memory location address</u>. Referring figs. 20-23, col. 23 lines 22-29, "switched channel memory controller" allows multiple DMA (and processors) to simultaneously communicate with multiple output channels. Also, col. 23, lines 40-45, suggests that CPU bus can be connected to an external flash memory through one channel and SDRAM through another channel. These statements clearly state there are separate and selective communication interfaces between the connections, figs. 21, 22 and 23 shows separate lines are connected to the memory interfaces). **(switch providing separate/distinct interfaces are known in the art, because a separate and independent interface avoids bus or memory bank conflict and hence increases the speed of the system [motivation to use switch], see Jeddeloh US 7,133,972, fig. 3, col. 4, lines 30-64, presented as an evidentiary reference).**

Zaidi explicitly fails to teach line cache receiving first address and comparing address to stored addresses and if match occurs it returns data to CPU, and retrieves data from one of the first and second memories if miss occurs, but a system with processor and cache memory is well known to one of ordinary skill in the art at the time of invention, and when CPU issues read request in such a system, cache compares the address with the stored addresses and returns the data to CPU and if miss occurs it retrieves data from higher latency storages (Jim, page 42-43, section 2.1.3) (also applicant's admitted prior art in the background of the invention section).

Zaidi teaches sending first address with bank (memory) select portion and memory location address (Zaidi, col. 23, lines 31-34) but fails to teach sending a second address based on first address. Taylor teaches computer systems using physical cache, which requires address translation occurring before the cache access (cache receiving translated/second address based on first address) (Taylor, column 1, lines 26-40). It would have been obvious to one having ordinary skill in the art at the time of the invention to utilize physical cache as taught by Taylor in the system of Zaidi and Jim because virtual memory provides protection, large address space and physical cache memories are simpler to build (Taylor, column 1, lines 26-40).

Zaidi, Jim and Taylor explicitly fail to teach the limitation "wherein said switch includes a plurality of selectors that each receives the second address and each select between first and second sets of signals relating to the first and second memory devices, respectively based on the second address". However, Zaidi teaches CPUs connect to either SRAM or flash memory through switched channel memory controller

as explained above, and the selection of respective signals are inherent in the system of Zaidi, because it is well known at the time of the invention to one having ordinary skill in the art that the signals (such as clock and request and acknowledgement etc.) are required for proper communication to be occur between the processor and cache or between cache and lower level storage systems (**see Jeddeloh, col. 4, lines 8-29, "the memory hub (switch) includes a processor interface 150 that is coupled to the processor 104 through a plurality of bus and signal lines, as is well known in the art". Jeddeloh, col. 4, lines 56-67, "for example, the switch 160 may be a cross-bar switch that can simultaneously couple at the processor interface and the memory interfaces 170a-c to each other. The switch can also be a set of multiplexers (plurality of selectors)"**).

As per claims 2-3, 45-46 and 80-81, Zaidi teaches that the first memory device is RAM (fig. 1, item 108).

As per claims 4, 47 and 82, Zaidi teaches the second memory device is flash memory (fig. 1, item 106).

As per claim 5, 48 and 83, Zaidi teaches the first CPU is an advanced risc machine (ARM) processor (column 5, lines 35-36).

As per claims 11 and 89, Zaidi teaches a cache memory as per claim 1 and memories are used for storing data. Zaidi fails to teach cache with a Content Addressable Memory (CAM). Jim teaches a cache memory with CAM, which stores addresses associated with data stored in the cache memory (page 14, sec. 1.5, page 15, fig. 1.7). Jim teaches determining when hit and miss occurs and retrieves data from higher latency memories (first and second memories) when miss occurs (Jim, pages 42-

43, sec. 2.1.3 and pages 46-47, fig. 2.4). Thus Jim inherently teaches cache state machine.

It would have been obvious to one having ordinary skill in the art at the time of invention have used Zaidi's dual processor system and modified to use the cache with CAM as taught by Jim because CAM permits content of memory to be searched and matched instead of having to specify a memory location in order to retrieve data from memory (Jim, page 14, sec. 1.5). This allows data to be stored at any location in a cache (Jim, page 16, paragraph 3)

As per claims 13-14, 52-53 and 91-92, Jim teaches a cache replacement algorithm Least Used Page, which replaces least used page with data retrieved form the first or second memory when miss occurs (page 57, paragraph 2, page 61, pars. 3-4). Thus Jim inherently teaches least used page device.

As per claims 15 and 54 Jim teaches that state transitions of cache state machine are based, in part on at least one internal state of the CPU (page 42, paragraph 2 and 3 and sec. 2.1.3)

As per claims 20 and 97, Jim teaches that cache can have many ways of implementations depending upon the address bits used in the system (page 54, paragraphs 2 and 3). Thus Jim inherently teaches cache with 4 pages of 8 x 32.

9. Claims 16-18, 55-57 and 93-95 are rejected under **35 U.S.C. 103(a)** as being unpatentable in view of Zaidi, Jim Handy, Taylor and Jeddeloh and in further view of Bryant et al. (4,008,460).

Claims 16-18 are similar in scope with combination of claims 1, 11, 13 and 14. Zaidi, Jim Handy, Taylor and Jeddeloh teach all the limitations of claim 16, including identifying first least used page and replacing first least used page in case of cache miss (limitation of claim 17) but fail to teach identifying first and second used page and replacing second least used page (claim 18). Bryant teaches identifying first and second least used page and replacing second least used page (Bryant, col. 3, lines 52-57).

It would have been obvious to one having ordinary skill in the art at the time of the invention to utilized first and second least used page replacement method as taught by Bryant in the system of Zaidi, Jim Handy, Taylor and Jeddeloh to avoid wrap-around delay associated with LRU policy and increase system performance (Bryant, col. 2, lines 7-16, lines 43-46).

Claims 55-57 and 93-95 are rejected under same rationales as applied to claims 1, 11, 13-14 and 16-17.

10. Claims 6, 21-24, 28, 49, 59-63, 84, 98-101 and 105 are rejected under **35 U.S.C. 103(a)** as being unpatentable over Zaidi, Jim Handy, Taylor and Jeddeloh as applied to claims 1-5, 11, 13-15, 44-48 and 79-83 above, and further in view of Barroso et al. (US 6,725,334 B2).

As per claims 6, 49 and 84, Zaidi and Jim teach a dual processor system with two caches (fig. 2, items 202 and 214, first and second processors, and items 208 and 224, two caches). Zaidi, Jim and Taylor inherently teach cache interface with first and second CPUs and both generates read requests and hence first and second address as taught in claim 1. Zaidi teaches system with two second level caches for two processors but fails to teach cache arbitration device which communicates with first and second cache interfaces and resolves cache access conflicts between first and second CPUs. Barroso teaches a second level cache with switch (fig. 1, item 130 and 120), which provides interfaces with first and second CPU and arbitrates between first and second CPU (column 4, lines 10-21) (**switch providing separate/distinct interfaces are known in the art, because a separate and independent interface avoids bus or memory bank conflict and hence increases the speed of the system, see Jeddeloh US 7,133,972, fig. 3, col. 4, lines 30-64, presented as an evidentiary reference**).

It would have been obvious to one having ordinary skill in the art at the time of invention would have modified the multiple cache with multiple processor system of Zaidi and used one cache with switch as taught by Barroso to reduce the cost and the waste of the cache capacity (column 1, lines 45-65).

Claims 21-24, 59-63, 84, 98-101 and 105 are similar in scope with combination of claims 1-6, 11, 13-15 and hence rejected under same rationales as applied to claims 1-6, 11 and 13-15 above.

11. <u>Claims 7-10, 25-27, 85-88 and 102-104 are rejected under **35 U.S.C. 103(a)** as being unpatentable over Zaidi, Jim Handy, Taylor, Jeddeloh and Barroso as applied to</u>

<u>claims 1, 6, 11, 13-14, 16-17, 44-48 and 79-84 above, and further in view of Alexander et al. (6,131,155).</u>

As per claims 7-8, Zaidi, Jim Handy, Taylor, Jeddeloh and Barroso teach all the limitations of claims 1-6 above but fail to teach direct interfaces from CPUs to memory devices. Alexander teaches CPU programmed to accessing main memory directly, bypassing cache access (Alexander, abstract).

It would have been obvious to one having ordinary skill in the art at the time of the invention to utilize the direct access interface to memory, bypassing the cache as taught by Alexander in the system of Zaidi, Jim Handy, Taylor, Jeddeloh and Barroso, because data caches provides performance improvement only if program execution performs repeated accesses of data over a short period of time to a small group of data and large amounts of data transfers degrades the performance, so bypassing a cache and directly reading data from memory increases the performance (Alexander, abstract, col. 2, lines 21-56). Also providing a direct and independent interface avoids bus or memory bank conflict as explained with respect to claims 1 and 6 above.

As per claim 9, Zaidi and Barroso teach an arbiter and MAC (Zaidi, fig. 2, items 242, 244) and switch (Barroso, fig.1, item 120) to resolve memory access conflict (Zaidi, column 23, lines 40-45) but fail to teach arbiter for direct read/write interface. It would have been obvious to one having ordinary skill in the art at the time of the invention would provide arbiter for direct (bypassing cache interface) interface, because when multiple CPUs accessing memory device providing arbitration avoids the conflict for same data.

As per claim 10, Zaidi teach an application specific integrated circuit (ASIC) which can be used to provide interconnection structure and method for efficient integration variety functional circuits (Zaidi, column 2, lines 63-65). It would have been obvious to one having ordinary skill in the art at the time of invention have used the embedded system of Zaidi to control the hard disk drive and its components for better performance and compact design.

Claims 25-27, 85-88 and 102-104 are rejected under same rationale as applied to claims 7-10 as above.

12. <u>Claims 19, 58 and 96 are rejected under **35 U.S.C. 103(a)** as being unpatentable over Zaidi, Jim Hardy, Taylor, Jeddeloh and Barroso and further in view of Veidenbaum et al. (Adapting Cache Line Size to Application Behavior, pub. 1999).</u>

Claims 19, 58 and 96 are similar in scope with combination of claims 1-6 above. But the combination of Zaidi, Jim, Taylor, Jeddeloh and Barroso fail to teach selecting size of the cache line based on application running. Veidenbaum teaches adapting cache line size according to application running (Veidenbaum, abstract).

It would have been obvious to one having ordinary skill in the art at the time of the invention to use cache line size based on application running as taught by Veidenbaum in the system of Zaidi, Jim, Taylor, Jeddeloh and Barroso to improve miss rate and memory traffic (Veidenbaum, abstract).

13. Claims 12, 51 and 90 are rejected under **35 U.S.C. 103(a)** as being unpatentable over Zaidi, Jim Hardy, Taylor, Jeddeloh and Barroso and further in view of Ebner et al. (US 6,928,525).

Claim 12 is similar in scope with combination of claims 1-6 and Zaidi, Jim Hardy, Taylor, Jeddeloh and Barroso teach all the limitations, but they combined failed to teach accessing one page by one of first and second CPUs, while other of first and second CPUs is accessing another page. Ebner teaches shared cache memory, which allows multiple simultaneous access to data held in different cache lines of cache (Ebner, Abstract). It would have been obvious to one having ordinary skill in the art at the time of the invention to utilize shared cache allowing multiple simultaneous access to different lines of cache as taught by Ebner in the system of Zaidi, Jim Hardy, Taylor, Jeddeloh and Barroso to improve system performance by allowing concurrent accesses to cache lines (Ebner, col. 2, lines 32-39).

Claims 51 and 90 are also rejected under same rationales as applied to claim 12.

Conclusion

14. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kaushikkumar Patel whose telephone number is 571-272-5536. The examiner can normally be reached on 8.00 am - 4.30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kaushikkumar Patel
Examiner
Art Unit 2188

kmp

HYUNG S...
SUPERVISORY PAT...
4-24-07

FORM HDP-1449 (Based on Form PTO-1449) **PATENT AND TRADEMARK OFFICE** **INFORMATION DISCLOSURE CITATION** (Use several sheets if necessary) Sheet 1 of 1	ATTORNEY DOCKET NO. MP0390	SERIAL NO. 10/626,507
	APPLICANT Son Ho et al	
	FILING DATE 7/24/2003	GROUP 2188

U.S. PATENT DOCUMENTS

Ref. Desig.	Examiner's Initials	Document Number	Date	Name	Class/ Subclass	(If appropriate) Filing Date
1.	/KP/	2005/0021916 A1	1/2005	Loafman, Zachary M.		
2.	/KP/	4,425,615	1/1984	Swenson et al		
3.	/KP/	6,725,339 B2	4/2004	Fu et al		

FOREIGN PATENT DOCUMENTS

Ref. Desig.	Examiner's Initials	Document Number	Date	Country	Class/ Subclass	Translation Yes	No
1.							

OTHER DOCUMENTS (including Author, Title, Date, Pertinent Pages, etc.)

Ref. Desig.	Examiner's Initials	
1.		

Examiner: /Kaushikkumar Patel/ (04/27/2007)	Date Considered:

EXAMINER: Please initial if citation considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.